Accurate Scene Text Recognition with Efficient Model Scaling and Cloze Self-Distillation

Supplementary Material

7. Context Update in Permutation Language Decoder

In Sec. 4, we introduced the architecture of the Permutation Language Decoder (PLD) used in our STR model. Specifically, in our implementation, each block of PLD receives the output of the previous block as the input of the query stream, while the key-value stream is provided with the same context and vision tokens across all blocks. It simplifies the original approach used in PARSeq [3], which updates the context when multiple blocks are presented. While the positional queries in PARSeq follow the same query stream as in our implementation, PARSeq additionally provides the context as input to the query stream in a second forward pass. This is done in order to update it before using it as input of the key-value stream of the next block. Fig. 6 shows the diagram of PLD in the PARSeq implementation: the positional queries follow the same path as in our implementation (black arrows), the context is updated following the red arrows.

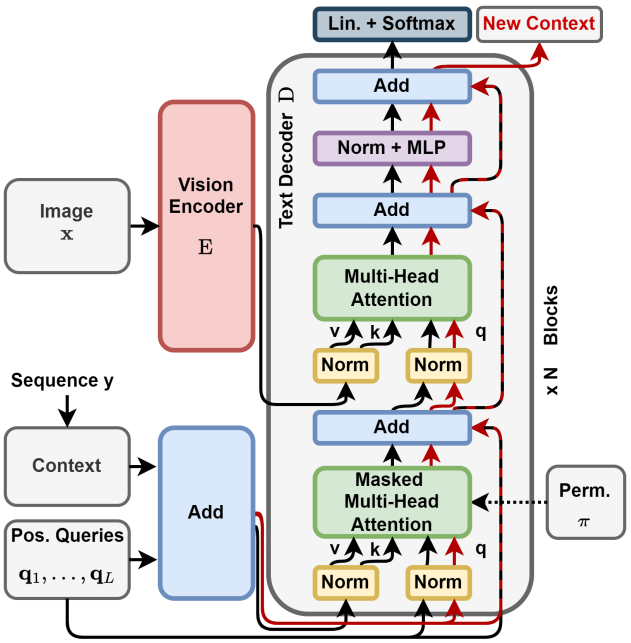


Figure 6. **Diagram of the context update in PLD.** The positional queries are updated following the black arrows (as in our Permutation Language Decoder). The context is updated following the red arrows: it is used as input of the query stream in a second forward pass before using it in the following block.

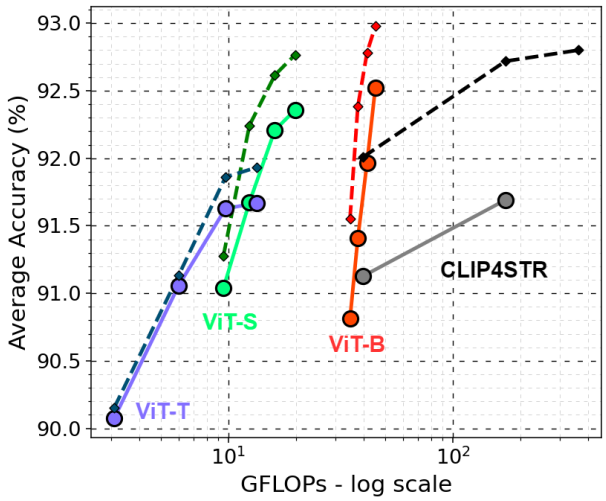


Figure 7. **Average word accuracy** (%) on 11 STR benchmarks for the models with ViT-T, ViT-S and ViT-B vision encoders and 4 different decoder sizes (see Sec. 4.1). Results are compared with the previous state-of-the-art model, CLIP4STR [54]. Results using **Real** training dataset (3.3M images) are depicted with solid lines and circle markers, while results using **RBU** training dataset (6.5M images) are shown with dashed lines and diamond markers. The x-axis represents the **total number GFLOPs** on a logarithmic scale.

Empirically, we found that this additional context update degrades the performance. Considering the average word accuracy across 11 benchmarks ($\mathbf{AVG}_{11}$), the performance of ViT-Base with PLD-Base decreases 0.15%, while ViT-Small and PLD-Base have a decrease of 0.19%. Moreover, since the context is also updated, the computational complexity is also increased. To this end, in all our analyses and experiments, we do not update the context as a default setting.

Remark. In PARSeq paper, they present the results using a single-block decoder so the context is actually not updated. However, their official implementation updates the context when multiple blocks are used.

8. Computational efficiency

In our STR model, the encoder presents a fixed computational cost, as it processes the vision tokens in a single forward pass. In contrast, the computational cost of the decoder depends on the sequence length due to the use of autoregressive (AR) decoding, which has been shown to outperform non-autoregressive (NAR) methods [3]. In Sec. 5.3,

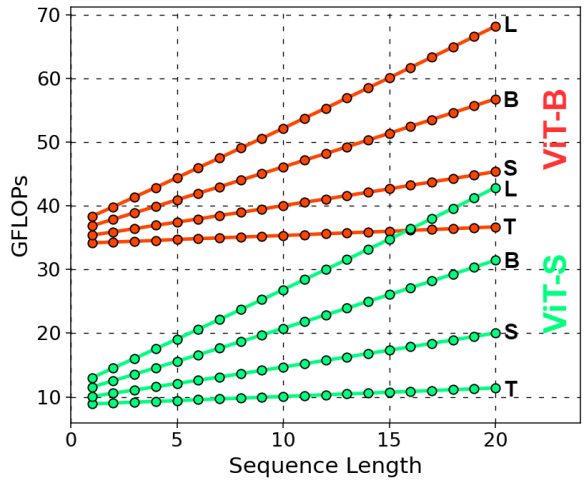


Figure 8. **GFLOPs for different sequence lengths.** The x-axis represents the sequence length (from 1 to 20 characters), while the y-axis represents the number of GFLOPs. Results are reported for ViT-Base and ViT-Small encoders paired with different decoders (PLD-T, PLD-S, PLD-B and PLD-L).

we demonstrated that increasing the decoder size is effective to improve performance. In this section, we analyze the impact of decoder size on overall GFLOPs.

Fig. 7 illustrates how the average model accuracies and the GLOPs change together. A similar plot is provided in Fig. 1 for the average model accuracy and the total number of parameters. The GFLOPs are calculated based on the average sequence length of 5.5, which corresponds to the average sequence length across all benchmark datasets. The plot reveals a similar trend that is observed for the number of parameters. Additionally, Fig. 8 shows how GFLOPs vary across different sequence lengths (from 3 to 20 characters) for various decoder sizes using ViT-B and ViT-S as encoders. Notably, for short sequence lengths, the encoder has the highest computational cost compared to the decoder. However, as the sequence length increases, decoder's GFLOPs increase, particularly for larger decoders. In most STR tasks, efficiency for long sequences is not a primary target since this kind of sequences is less common in natural scene settings.

Remark. When referring to the sequence length, we specifically consider the number of characters to be decoded. In the actual implementation, two additional special tokens are also decoded: the beginning-of-sequence token (BOS) and the end-of-sequence token (EOS), which mark the start and end of decoding process, respectively. The computation of these tokens is included in the GFLOPs calculation for any sequence length.

9. Cloze Self-Distillation: hyperparameters

In Sec. 4.2, we introduced Cloze Self-Distillation, our novel technique to train STR models on real data. The objective of CSD is presented in Eq. 10 that we report here for convenience:

$$\min_{\theta} \mathbb{E}_{\substack{(\mathbf{x},\mathbf{y})\sim\mathcal{D} \\ \boldsymbol{\pi}\sim\Pi \\ t\sim[1,L]}} \left[-\log p_\theta(y_{\pi_t} | \mathbf{y}_{\boldsymbol{\pi}_{<t}}, \mathbf{x}) + \alpha \mathrm{KD}_{\boldsymbol{\pi},t}(\mathbf{x}, \mathbf{y}) \right]$$

In the experiments presented in the main text, we set the mixing hyperparameter and distillation temperature to $\alpha = 0.1$ and $\tau = 2.0$, respectively. In this section, we present a post-hoc ablation study to show that CSD is not highly sensitive to these hyperparameters. To provide consistent results for different values of α without changing the learning rate and training dynamics, in this section, we multiply the loss by $\frac{1+\alpha_0}{1+\alpha}$, where $\alpha_0 = 0.1$ is our base value for α. Tab. 7 shows that for each combination of α and τ within the considered range, the average word accuracy of CSD surpasses both the accuracy achieved using solely pseudolabels (92.3%) and the accuracy obtained through conventional training methods (92.0%). Moreover, increasing the temperature and mixing parameter appears to further enhance performance beyond the results presented in the main text.

	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1.0$
$\tau = 1.0$	92.4	92.4	92.5
$\tau = 2.0$	92.5	92.5	92.5
$\tau = 3.0$	92.5	**92.6**	**92.6**

Table 7. **CSD hyperpameters.** Average word accuracy (%) $\mathbf{AVG}_{11}$ using CSD-B (ViT-Base + PLD-Base) with Real dataset for different values of mixing parameter α and temperature τ.

10. Architecture Analysis

In Sec. 4.3, we presented our Permutation Language Decoder equipped with Differential Cross-Attention layers. The aim was to minimize the amount of noise present in the attention maps. Tables 8, 9 and 10 provide visual comparisons between the standard Cross-Attention and our Differential Cross-Attention. From the results, the majority of the noise and errors observed in the standard Cross-Attention are effectively reduced when the Differential Cross-Attention is used.

11. Effectiveness of CSD

All the components of CSD, namely pseudo-labels, knowledge distillation of the context-aware predictions and differential decoder, provide substantial improvements as presented in Tab. 3 and Tab. 4. Specifically, on Real dataset with the base model, pseudo-labels (PL) provide +0.26% improvement by themselves. When PL and context-aware KD are combined, the improvement is +0.50% (providing robustness to label noise). Finally, the differential decoder

(DD) provides an additional relevant improvement (mitigating attention noise): PL + KD + DD obtains $+0.70\%$. Notably, many benchmarks in STR (used to compute the average accuracy) are saturated and affected by test label errors. For this reason, while the improvements might seem modest, they are significant. For comparison, CLIP4STR scales the architecture from 158M to 446M parameters to obtain only $+0.56\%$ improvement.

12. Additional Results

In Table 11 and 12, we present qualitative examples of predictions of our STR model by comparing with CLIP4STR [54]. From the results, even if CLIP4STR has a separate branch for text correction, our STR model obtains more accurate results, especially for occluded cases. This shows that training encoder-decoder parts together provides robustness and improves the accuracy. In Table 13 we show that CSD outperforms previous state-of-the-art even in the challenging Union14M benchmark.

Table 8. **Comparison of Attention Maps.** Attention maps of the last Cross-Attention in the last block of the Permutation Language Decoder. On the left: the original input image. First row of each section: attention maps obtained with the standard Cross-Attention. Second row of each section: attention maps obtained with our Differential Cross-Attention.

Table 9. **Comparison of Attention Maps.** Attention maps of the last Cross-Attention in the last block of the Permutation Language Decoder. On the left: the original input image. First row of each section: attention maps obtained with the standard Cross-Attention. Second row of each section: attention maps obtained with our Differential Cross-Attention.

Table 10. **Comparison of Attention Maps.** Attention maps of the last Cross-Attention in the last block of the Permutation Language Decoder. On the left: the original input image. First row of each section: attention maps obtained with the standard Cross-Attention. Second row of each section: attention maps obtained with our Differential Cross-Attention.

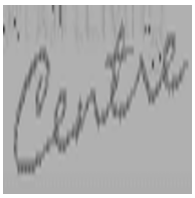
Image	Ground Truth	CLIP4STR-L	CSD-D (ours)	Image	Ground Truth	CLIP4STR-L	CSD-D (ours)
	260	250	260		cheuvront	cheu_ront	cheuvront
	3123410900	3113410900	3123410900		electric	electnic	electric
	arlboro	arljoro	arlioro		cottages	cottages	cottagee
	assistence	wassistence	assistence		cotton	cutton	cutton
	bubble	bubble	bibble		haircut	hai_cut	haircut
	capogiro	cap_giro	capogiro		hotel	lotel	hotel
	centre	centie	centre		kaffee	laffee	kaffee

Table 11. **Qualitative examples.** The table presents image examples along with ground truth labels, predictions made by our models, CSD-D and CLIP4STR-L, which were both trained using the RBU dataset. These predictions are based on a character set consisting of 36 alphanumeric characters. Errors are highlighted in red.

Image	Ground Truth	CLIP4STR-L	CSD-D (ours)	Image	Ground Truth	CLIP4STR-L	CSD-D (ours)
	kennedy	kenned_	kennedy		tabu	tqbu	tabu
	lower	power	_ower		three	thpee	three
	northeast	nertheast	northeast		tigger	tiggen	tigger
	menuboard	menuboard	meruboard		towe_	tower	tower
	loccasio	loccasio	loccasio		valerie	valerte	valerie
	scientific	scientifi_	scentifi_		vigilant	vigitant	vigitant
	spaghetti	spachetti	spaghetti		immortals	immortals	jmmortals

Table 12. **Qualitative examples.** The table presents image examples along with ground truth labels, predictions made by our models, CSD-D and CLIP4STR-L, which were both trained using the RBU dataset. These predictions are based on a character set consisting of 36 alphanumeric characters. Errors are highlighted in red.

Method	Data	Params	Curve	Multi-Oriented	Artistic	Contextless	Salient	Multi-Words	General	Avg
CLIP4STR-B	Real	158M	96.3	96.1	86.5	**92.2**	91.2	88.9	89.9	91.6
CLIP4STR-L	Real	446M	**97.0**	96.6	87.2	91.0	91.5	**89.9**	90.3	91.9
CSD-D (ours)	Real	**110M**	**97.0**	**97.0**	**87.7**	91.8	**91.7**	89.5	**91.7**	**92.3**
CLIP4STR-B	REBU-Syn	158M	96.4	96.3	**88.6**	90.1	91.9	**92.2**	89.1	92.1
CLIP4STR-L	REBU-Syn	446M	96.4	**97.2**	**88.6**	90.4	92.7	90.7	89.3	92.2
CSD-D (ours)	RBU	**110M**	**96.5**	**97.2**	**88.6**	**92.8**	**92.8**	90.8	**90.2**	**92.7**

Table 13. Comparison of CSD-D with CLIP4STR (results from [31]) on Union14M benchmark.